Modelowanie i Analiza Systemów Informatycznych

Logika Temporalna i Automaty Czasowe

(4)

Modelowa weryfikacja systemu

© Paweł Głuchowski, Politechnika Wrocławska

wersja 2.1

Treść wykładu

Własności współbieżnych programów

- Co to jest współbieżny program?
  - Typowe własności

Treść wykładu

Automat skończenie stanowy

- Co to jest automat skończenie stanowy?
- Przykłady automatów i ich własności

Treść wykładu

Modelowa weryfikacja systemu

- Modelowa weryfikacja systemu
- Najważniejsze własności do weryfikacji
- Algorytm modelowej weryfikacji systemu
- Prosty przykład

Własności współbieżnych programów

- Co to jest współbieżny program?
  - Typowe własności

Własności współbieżnych programów

Co to jest współbieżny program?

- Współbieżny program składa się z procesów, gdzie:
  - procesy wykonywane są w tym samym czasie;
  - procesy mogą współdzielić pewne zasoby, np. zmienne;
  - procesy mogą wzajemnie oddziaływać na siebie.

Własności współbieżnych programów

Typowe własności

- Współbieżny program P składa się ze współbieżnie wykonywanych procesów: P1, P2, P3, …
- Własności P to: q_1, q_2, q_3, q_4, …
- Własność q_i programu P może być określona przez formułę ϕ_i, która będzie spełniona w każdym wykonaniu P (od pierwszego stanu, w którym $s_0 \models \phi_i$), niezależnie od sposobu zarządzania procesami.

Własności współbieżnych programów

Typowe własności

- **bezpieczeństwo** – q_1 jest spełnione <u>w każdym momencie</u>

  $\phi_1 \equiv Gq_1$ np. dostęp do sekcji krytycznej CS zawsze jest możliwy.

- **osiągalność** – q_2 będzie spełnione <u>w pewnym momencie</u>

  $\phi_2 \equiv Fq_2$ np. proces w końcu dostanie się do CS.

- **odpowiedź** – q_3 jest spełnione <u>od czasu do czasu</u>

  $\phi_3 \equiv GFq_3$ np. proces jest usypiany od czasu do czasu.

- **trwałość** – <u>od pewnego momentu</u>:
  q_4 jest spełnione <u>w każdym momencie</u>

  $\phi_4 \equiv FGq_4$ np. od pewnego momentu: proces nie chce dostępu do CS.

- **żywotność** – q_5 jest osiągalne <u>jako skutek</u> p

  $\phi_5 \equiv G(p \Rightarrow Fq_5)$ np. dostęp procesu do CS jest zapewniony na żądanie.

Automat skończenie stanowy

- Co to jest automat skończenie stanowy?
- Przykłady automatów i ich własności

Automat skończenie stanowy

Co to jest automat skończenie stanowy?

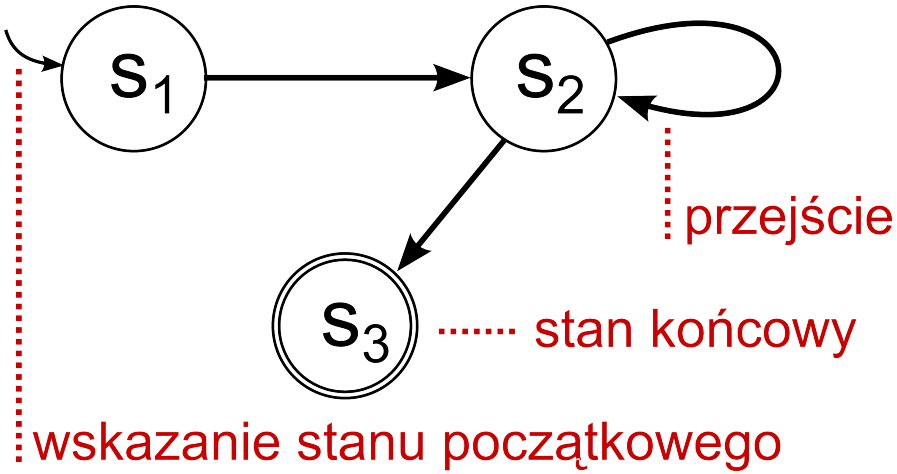
- jest abstrakcyjną maszyną stanową,
- składa się ze skończonej liczby stanów i przejść między nimi,
- ma stany początkowe i może mieć stany końcowe,
- jest automatem, w którym przejścia między stanami są odpalane deterministycznie
  (jednoznacznie opisane funkcją przejścia).

Automat skończenie stanowy

Co to jest automat skończenie stanowy?

Automat $A=\{\Sigma, S, S_0, \rho, F\}$ jest zdefiniowany przez:

- Σ – alfabet (zbiór stanów programu),
- S – zbiór stanów automatu,
- S_0 – zbiór stanów początkowych,
- ρ – funkcja przejścia ($S \times \Sigma \times S$),
- F – zbiór stanów końcowych.



Przebieg to ścieżka stanów, np. s_1, s_2, s_2, s_3.

Stan można opisać wyrażeniami, które w nim są prawdziwe.

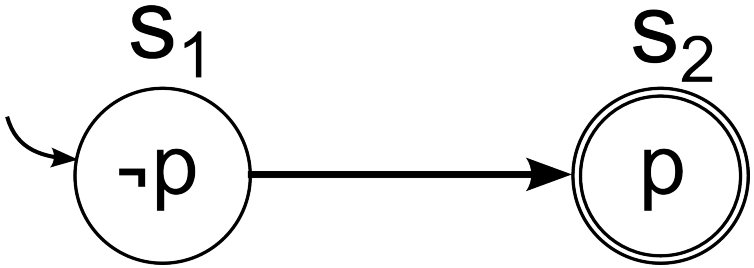
Automat skończenie stanowy

Przykłady automatów i ich własności

Automat zamykający drzwi

- Cel: automatycznie zamknąć drzwi, gdy są otwarte.
- Założenia:
  - na początku drzwi są otwarte,
  - drzwi zamykają się natychmiast (czas zamykania wynosi zero).
- Wyrażenie *p* — drzwi są zamknięte.
- W momencie (s_1) włączenia automatu prawdą jest, że:

  $\neg p \wedge XGp$

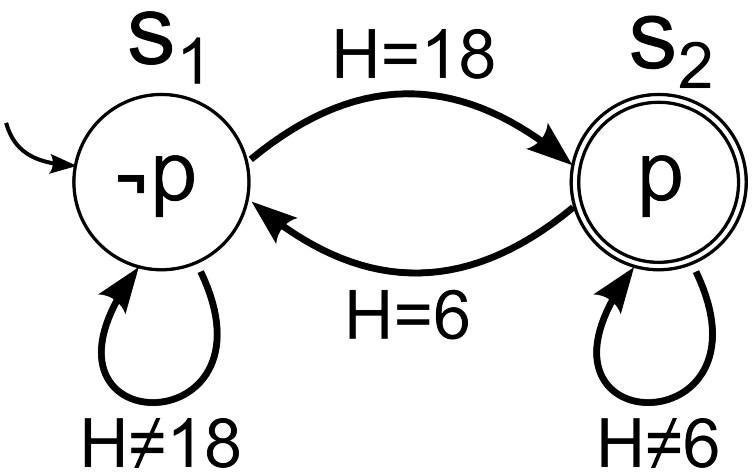


Automat skończenie stanowy

Przykłady automatów i ich własności

Automat włączający i wyłączający światło

- Cel: automatycznie włączyć światło o 18^{00} i wyłączyć o 6^{00}.
- Założenia:
  - na początku światło jest wyłączone,
  - światło włączane jest o 18^{00} i wyłączane o 6^{00},
  - włączanie i wyłączanie światła jest natychmiastowe,
  - zegar H startuje o 12^{00}.
- Wyrażenie *p* — światło jest włączone.



Automat skończenie stanowy

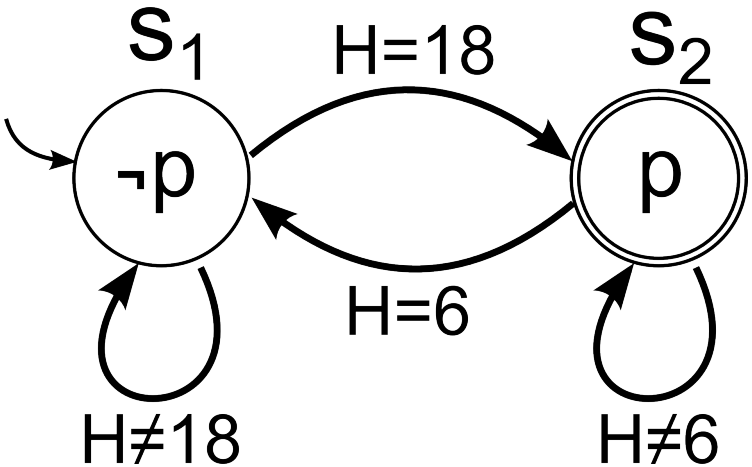
Przykłady automatów i ich własności

Automat włączający i wyłączający światło

- W momencie (s_1) włączenia automatu prawdą jest, że:

$$\neg p \wedge H{=}12 \wedge G(\neg p \wedge H{\neq}18 \Rightarrow X\neg p) \wedge G(\neg p \wedge H{=}18 \Rightarrow Xp) \wedge G(p \wedge H{\neq}6 \Rightarrow Xp) \wedge G(p \wedge H{=}6 \Rightarrow X\neg p)$$

- Automat spełnia poniższe własności:
  - lampa działa:
    $GFp \wedge GF\neg p$
  - lampa nie mruga:
    $\neg F(p \wedge X\neg p \wedge XXp)$

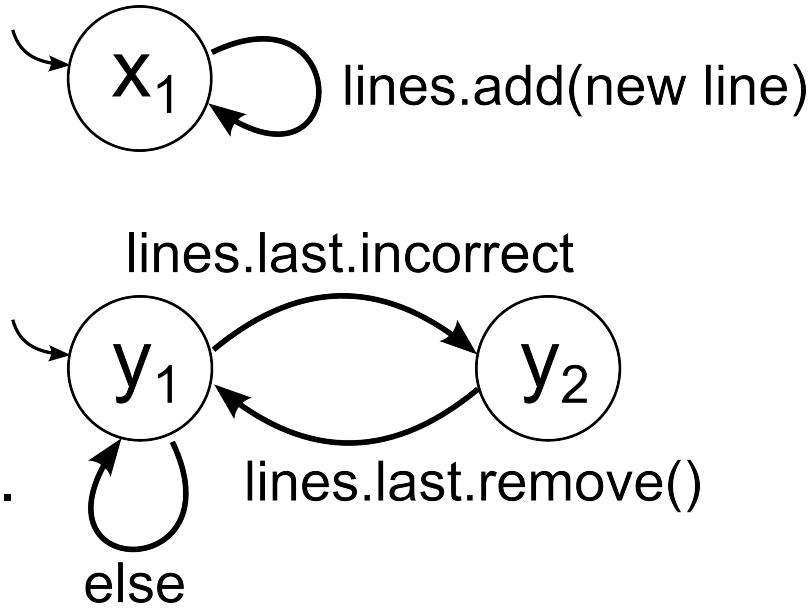


Automat skończenie stanowy

Przykłady automatów i ich własności

Automaty współbieżnych procesów

- Proces **P1** umieszcza linie tekstu w tabeli „*lines*”.
- Proces **P2** usuwa linie z „*lines*”, jeśli są niepoprawne.
- P1 i P2 wykonywane są współbieżnie.



P1: *while (true) do { lines.add(new line); }*

P2: *while (true) do { if(lines.last.incorrect) {lines.last.remove(); }}*

Automat skończenie stanowy

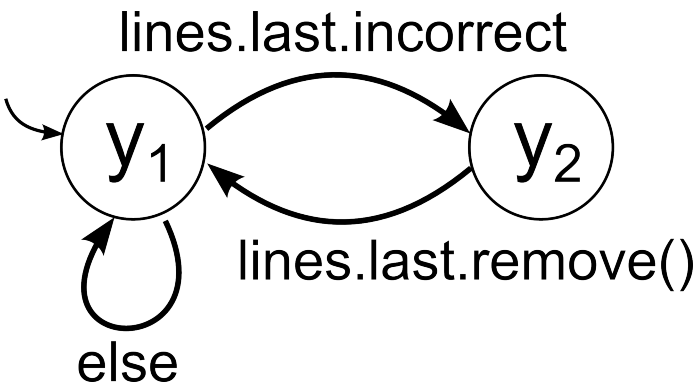
Przykłady automatów i ich własności

Automaty współbieżnych procesów

Czy te własności są spełnione?

- „*lines*” nigdy nie zawiera więcej niż jednej niepoprawnej linii:

  $\neg F(\text{lines.incorrect.count} > 1)$



- każda linia istnieje w „lines” przez jakiś czas:

  $(\forall \text{ line})(F(\text{lines.contains(line)}))$

Automat skończenie stanowy

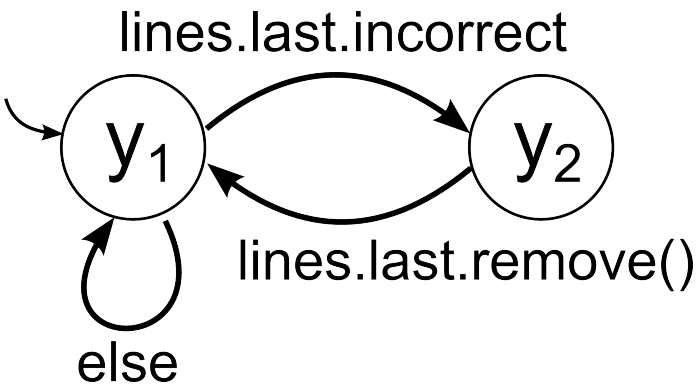
Przykłady automatów i ich własności

Automaty współbieżnych procesów

Czy te własności są spełnione?

- „*lines*” nigdy nie zawiera więcej niż jednej niepoprawnej linii:

  ¬F(lines.incorrect.count > 1) NIE



- każda linia istnieje w „lines” przez jakiś czas:

  (∀ line)(F(lines.contains(line))) TAK

Modelowa weryfikacja systemu

- Modelowa weryfikacja systemu
- Najważniejsze własności do weryfikacji
- Algorytm modelowej weryfikacji systemu
- Prosty przykład

Modelowa weryfikacja systemu

Dane wejściowe:

- formalny model funkcji systemu dany jako automat;
- własności, które muszą być spełnione przez system, dane jako formuły logiki temporalnej.

Dane wyjściowe:

- odpowiedź, czy system spełnia te własności.

Czy każda możliwa sekwencja stanów systemu (programu) spełnia te własności?

Czy któraś możliwa sekwencja stanów systemu (programu) spełnia te własności?

Modelowa weryfikacja systemu

Najważniejsze własności do weryfikacji

- **osiągalność**
  LTL: $\mathrm{F}p$
  CTL: $\mathrm{EF}p$
  – „pożądany” stan *p* systemu <u>w końcu</u> zostanie osiągnięty,
- **bezpieczeństwo**
  LTL: $\mathrm{G}\neg q$
  CTL: $\mathrm{AG}\neg q$
  – „niechciany” stan *q* systemu <u>nigdy</u> nie zostanie osiągnięty.

Cel: zweryfikuj (automatycznie) czy dana formuła jest prawdziwa w danym modelu.

Modelowa weryfikacja systemu

Algorytm modelowej weryfikacji systemu

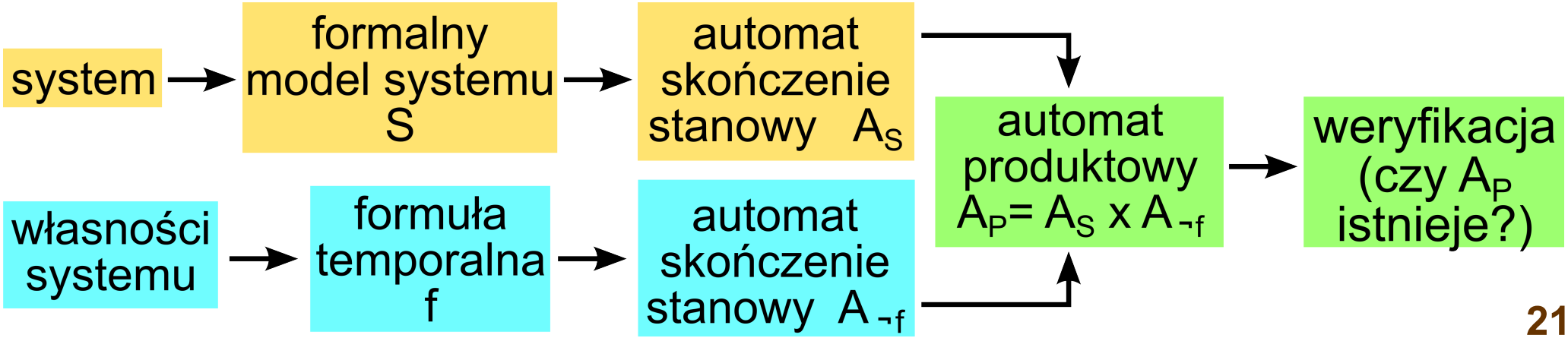
1) Zbuduj skończenie stanowy automat A_S dla modelu systemu S.

2) Zapisz własności systemu jako formułę f.

3) Zbuduj skończenie stanowy automat $A_{\neg f}$ dla formuły $\neg f$.

4) Zbuduj automat produktowy dla $A_P = A_S \times A_{\neg f}$.

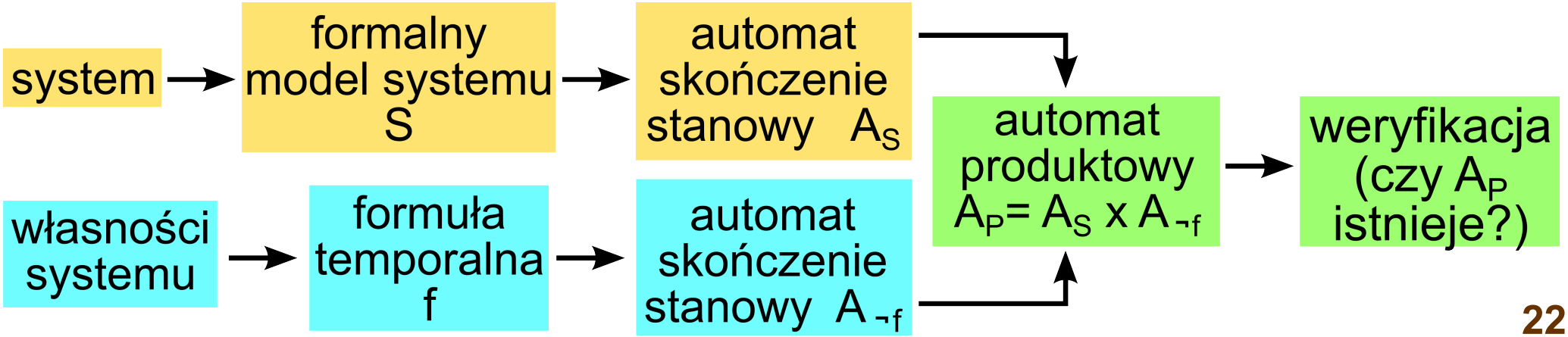
5) Zweryfikuj (automatycznie) czy A_P istnieje.



Modelowa weryfikacja systemu

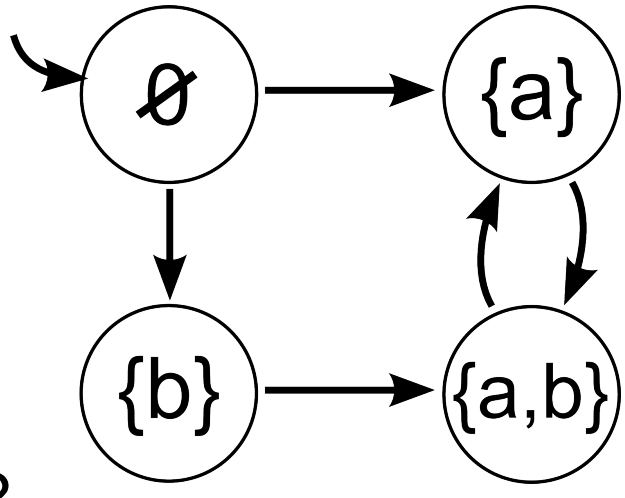
Algorytm modelowej weryfikacji systemu

- $A_{\neg f}$ powinien akceptować tylko takie sekwencje stanów programu systemu S, które spełniają formułę $\neg f$.
- Jeśli istnieje taka sekwencja stanów w A_S, która odpowiada formule $\neg f$, to możliwe jest takie wykonanie programu, które nie spełnia podanych własności.

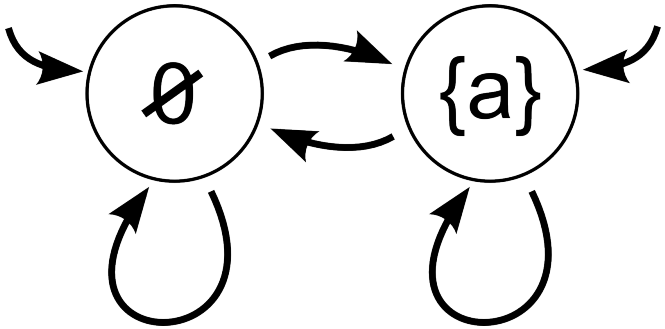


Modelowa weryfikacja systemu

Prosty przykład



- Automat A_S modeluje system, gdzie każdy stan etykietowany jest tymi wszystkimi wyrażeniami, które są w nim prawdziwe: *a* i *b*. (∅ – *a* i *b* są fałszywe)
- Czy własność $f \equiv EF b$ jest spełniona dla A_S?



- Automat $A_{\neg f}$ modeluje formułę $\neg f \equiv \neg EF b \equiv AG \neg b$.
- W A_S nie istnieje żaden nieskończony przebieg $A_{\neg f}$ – własność f jest spełniona.

Zauważ, że własność AF*b* *też jest spełniona.*

Koniec

Literatura:

- R. Alur et al. „Automata for modelling real-time systems”, 1990
- J.E. Hopcroft, J.D. Ullman „Introduction of Automata Theory, Languages, and Computation”, 2001
- R. Klimek *„Zastosowanie logiki temporalnej w specyfikacji i weryfikacji oprogramowania – w stronę czasu rzeczywistego”*, w: „Wykłady zaproszone. Systemy Czasu Rzeczywistego 2000”, 2001